B : câu hỏi

Câu 1 :

-\*p3 = B , p3 = 0x5678;

-\*p3 = 0x1234 , p3 = A ;

-\*p1 = 0x5678 , p1 = B ;

Câu 2 :

K = 5 và \*p = 5 ;

Câu 3 :

Lỗi vì c đag có kiểu char nếu muốn làm như vậy phải ép kiểu ;

C : bài tập

int dodai(const char \*s) {

int length = 0;

while (\*s++) {

length++;

}

return length;

}

1 . hàm đảo ngược chuỗi ;

void reverse(char \*a) {

int len = my\_strlen(a);

char \*start = a;

char \*end = a + len - 1;

while (start < end) {

char temp = \*start;

\*start = \*end;

\*end = temp;

start++;

end--;

}

}

2 . hàm xóa kí tự c trong xâu ;

void delete\_char(char \*a, char c) {

char \*src = a, \*dst = a;

while (\*src) {

if (\*src != c) {

\*dst++ = \*src;

}

src++;

}

\*dst = '\0';

}

3 . độn chuỗi để có độ dài chuỗi a bằng độ dài = n ;

void pad\_right(char \*a, int n) {

int len = dodai(a);

if (len >= n) return;

char \*end = a + len;

for (int i = len; i < n; i++) {

\*end++ = ' ';

}

\*end = '\0';

}

4 . độn trái để ...

void pad\_left(char \*a, int n) {

int len = dodai(a);

if (len >= n) return;

int shift = n - len;

for (int i = len; i >= 0; i--) {

a[i + shift] = a[i];

}

for (int i = 0; i < shift; i++) {

a[i] = ' ';

}

}

5 . cắt xâu ;

void truncate(char \*a, int n) {

if (dodai(a) > n) {

a[n] = '\0';

}

}

6 . đối gương ;

bool is\_palindrome(const char \*a) {

int len = dodai(a);

const char \*start = a;

const char \*end = a + len - 1;

while (start < end) {

if (\*start != \*end) {

return false;

}

start++;

end--;

}

return true;

}

7 . lọc trái ;

void trim\_left(char \*a) {

char \*start = a;

while (\*start == ' ') {

start++;

}

char \*dst = a;

while (\*start) {

\*dst++ = \*start++;

}

\*dst = '\0';

}

8 . lọc phải ;

void trim\_right(char \*a) {

int len = dodai(a);

if (len == 0) return;

char \*end = a + len - 1;

while (end >= a && \*end == ' ') {

end--;

}

\*(end + 1) = '\0';

}

D . cau hoi trac Nghiem ;

1 . \*p = n ;

2 . \*p1 = p2 ;

3 . ii & iii ;

4 . iii ;

5 . iv ;

6 . iii ;

7 . iii ;